Coordination-free Byzantine Replication with Minimal Communication Costs

Jelle Hellings Mohammad Sadoghi

Exploratory Systems Lab,
Department of Computer Science,
University of California, Davis, CA, USA

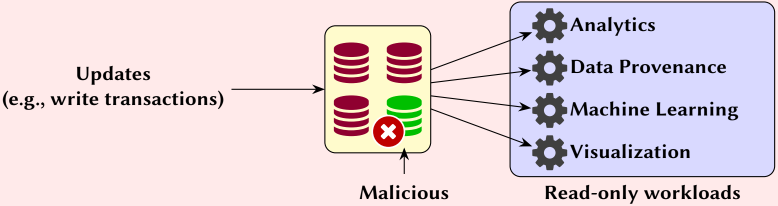
Towards high performance fault-tolerant data processing

Fault tolerance via full replication

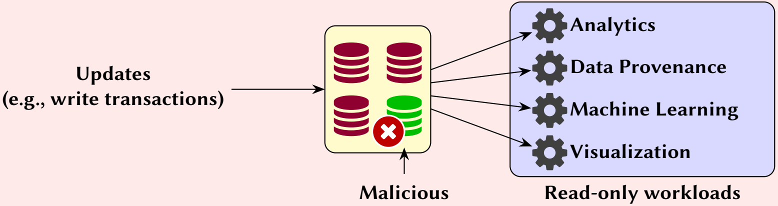
- *Communication intensive*: coordinate all steps of the system via *consensus*.
- *No scalability*: adding replicas reduces performance.
- *Replicas can be malicious*: read-only queries cannot be answered by single replicas.
- No sharding, no geo-aware data localization, no specialization.

Contradicts standard techniques used for high-performance data processing!

Towards our vision: specializing for read-only workloads



Towards our vision: specializing for read-only workloads



Enabling data-hungry read-only workloads

Need to stream all data updates with low cost for all replicas involved.

The need for Byzantine learning

Definition

Let $\mathfrak{R}$ be a cluster deciding on a sequence of transactions.
The *Byzantine learning problem* is the problem of sending the decided transactions from $\mathfrak{R}$ to a learner $\textsc{l}$ such that:

- the learner $\textsc{l}$ will eventually *receive all* decided transactions;
- the learner $\textsc{l}$ will *only receive* decided transactions.

Practical requirements

- Minimizing overall communication.
- Load balancing among all replicas in $\mathfrak{R}$.

Background: Information dispersal algorithms

Definition

Let v be a value with storage size $s = \|v\|$.
An *information dispersal algorithm* can encode v in $\mathbf{n}$ pieces v' such that v can be *decoded* from every set of $\mathbf{n} - \mathbf{f}$ such pieces.

Theorem (Rabin 1989)

The IDA algorithm is an optimal information dispersal algorithm:

- *Each piece v' has size* $\left\lceil \frac{\|v\|}{\mathbf{n}-\mathbf{f}} \right\rceil$.
- *The $\mathbf{n} - \mathbf{f}$ pieces necessary for decoding have a total size of* $(\mathbf{n} - \mathbf{f}) \left\lceil \frac{\|v\|}{(\mathbf{n}-\mathbf{f})} \right\rceil \approx \|v\|$.

The delayed-replication algorithm

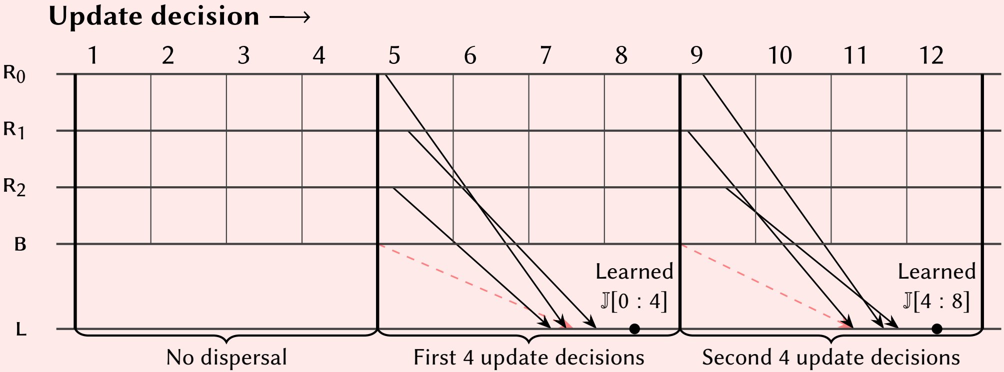
Idea: $\mathfrak{R}$ sends a journal to learner L

1. Partition the journal in sequences S of $\mathbf{n}$ transactions.
2. Replica $\mathrm{R}_i \in \mathfrak{R}$ encodes S into the i-th IDA piece S_i.
3. Replica $\mathrm{R}_i \in \mathfrak{R}$ sends S_i with a checksum $C_i(S)$ of S to L.
4. *L receives at least* $\mathbf{n} - \mathbf{f}$ *distinct pieces and decodes* S.

Observation ($\mathbf{n} > 2\mathbf{f}$)

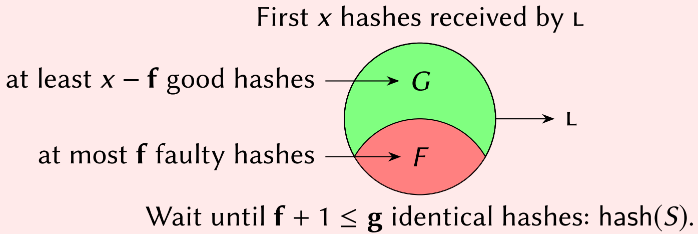
- Replica R_i sends at most $B = \left\lceil \frac{\|S\|}{\mathbf{n}-\mathbf{f}} \right\rceil + c \leq \frac{2\|S\|}{\mathbf{n}} + 1 + c = \mathcal{O}\left(\frac{\|S\|}{\mathbf{n}} + c\right)$ bytes.
- Learner L receives at most $\mathbf{n} \cdot B = \mathcal{O}\left(\|S\| + c\mathbf{n}\right)$ bytes.

Communication by the delayed-replication algorithm



Decoding S using simple checksums ($\mathbf{n} > 2\mathbf{f}$)

- Use checksums hash(S).
- The $\mathbf{n} - \mathbf{f}$ non-faulty replicas will provide correct *pieces*.
- At least $\mathbf{n} - \mathbf{f} > \mathbf{f}$ messages with correct *checksums*.



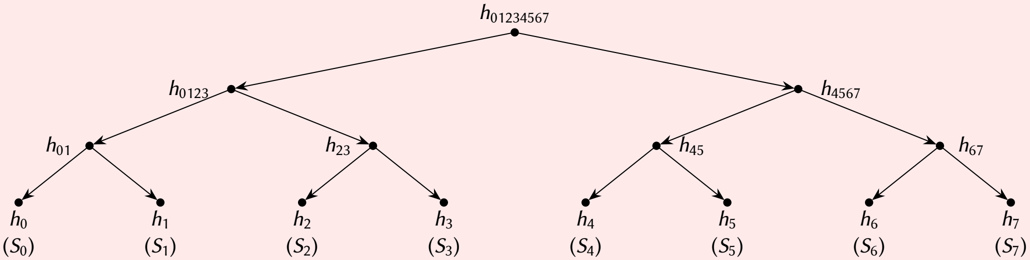
- Compute intensive for learners: one can choose $\mathbf{n} - \mathbf{f}$ out of $\mathbf{n}$ messages in $\binom{\mathbf{n}}{\mathbf{n}-\mathbf{f}}$ ways
  only one such choice is guaranteed to be correct!

Decoding S using tree checksums

Use Merkle-trees to construct checksums

Consider 8 replicas and a sequence S.

We construct the checksum $C_5(S)$ of S (used by $\textsc{r}_5$).



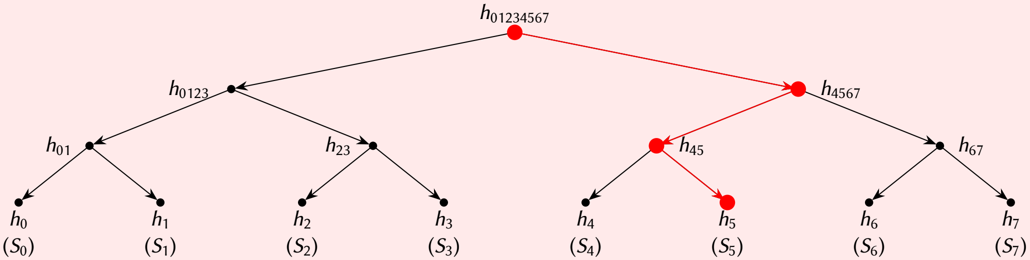
Construct a Merkle tree for pieces $S_0, \ldots, S_7$.

Decoding S using tree checksums

Use Merkle-trees to construct checksums

Consider 8 replicas and a sequence S.

We construct the checksum $C_5(S)$ of S (used by $\textsc{r}_5$).



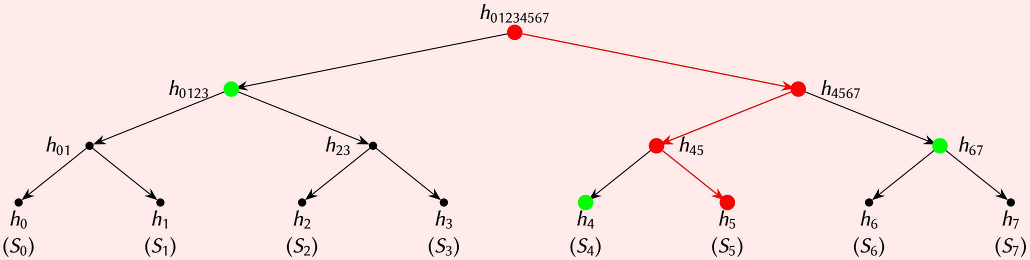
Determine the path from root to S_5.

Decoding S using tree checksums

Use Merkle-trees to construct checksums

Consider 8 replicas and a sequence S.

We construct the checksum $C_5(S)$ of S (used by $\textsc{r}_5$).



Select *root* and *neighbors*: $C_5(S) = [h_4, h_{67}, h_{0123}, h_{01234567}]$.

Delayed-replication: Main result ($\mathbf{n} > 2\mathbf{f}$)

Theorem

Consider the learner $\textsc{l}$*, replica* $\textsc{r}$*, and decided transactions* $\mathcal{T}$*. The delayed-replication algorithm with tree checksums guarantees*

1. $\textsc{l}$ *will learn* $\mathcal{T}$*;*
2. $\textsc{l}$ *will receive at most* $|\mathcal{T}|$ *messages with a total size of* $\mathcal{O}\left(\|\mathcal{T}\| + |\mathcal{T}| \log \mathbf{n}\right)$*;*
3. $\textsc{l}$ *will only need at most* $|\mathcal{T}|/\mathbf{n}$ *decode steps;*
4. $\textsc{r}$ *will sent at most* $|\mathcal{T}|/\mathbf{n}$ *messages to* $\textsc{l}$ *of size* $\mathcal{O}\left(\frac{\|\mathcal{T}\|+|\mathcal{T}|\log \mathbf{n}}{\mathbf{n}}\right)$.

Application: scalable storage for resilient systems

- Clusters typically need only a *view* $\mathbb{V}$ on the data to decide whether updates are valid.
- Clusters only need the full journal $\mathbb{J}$ for *recovery*.
- We can use *delayed-replication* to reduce the data each replica has to store.

Theorem

The storage cost per replica can be reduced from

$$\mathcal{O}\left(\|\mathbb{J}\| + \|\mathbb{V}\|\right) \quad \textit{to} \quad \mathcal{O}\left(\frac{\|\mathbb{J}\|}{\mathbf{n}-\mathbf{f}} + \frac{|\mathbb{J}|}{\mathbf{n}}\log(\mathbf{n}) + \|\mathbb{V}\|\right).$$

Conclusion

Efficient Byzantine learning is possible.

Blockchain applications

- Low-cost checkpoint protocols.
- Scalable storage for resilient systems.

Fault-tolerant high-performance data processing: ongoing work

- More at `https://jhellings.nl/` and `https://expolab.org/`.
- Paper: `https://doi.org/10.4230/LIPIcs.ICDT.2020.17`.